

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут»
Радіотехнічний факультет
Кафедра радіоконструювання та виробництва радіоапаратури

Автоматизація оброблення технічної інформації

Методичні вказівки до виконання лабораторних робіт
для студентів радіотехнічного факультету,
спеціальності 172 Телекомунікації та радіотехніка

Рекомендовано вченою радою радіотехнічного факультету

Київ
НТУУ «КПІ»
2016

Автоматизація оброблення технічної інформації : метод. вказівки до виконання лабораторних робіт для студентів радіотехнічного факультету спеціальності 172 Телекомунікації та радіотехніка / Уклад. : В. О. Адаменко — К. : НТУУ «КПІ», 2016. — 41 с.

Гриф надано вченою радою радіотехнічного факультету НТУУ «КПІ»
(Протокол № 05/2016 від 30 травня 2016)

Рекомендовано навчально-методичною комісією радіотехнічного факультету
(Протокол № 05/2016 від 27 травня 2016)

Навчальне видання

Автоматизація оброблення технічної інформації
Методичні вказівки до виконання лабораторних робіт
для студентів радіотехнічного факультету,
спеціальності 172 Телекомунікації та радіотехніка

Укладач: *Адаменко Володимир Олексійович, асистент*

Рецензент: *Сушко Ірина Олександрівна, к.т.н., доц.*

За редакцією укладача

ЗМІСТ

ВСТУП	4
Лабораторна робота №1 Елементарні функції та графічні побудови ...	5
Лабораторна робота №2 Робота з табличними даними	8
Лабораторна робота №3 Апроксимація табличних даних.....	11
Лабораторна робота №4 Розрахунок параметрів фільтру	15
Лабораторна робота №5 Фільтрація сигналів	19
Лабораторна робота №6 Обробка зображення	21
Лабораторна робота №7 Модуляція сигналів	26
Перелік посилань	33
Додаток А. Особливості роботи в MatLAB.....	34
Додаток Б. Приклад оформлення лабораторної роботи.....	39

ВСТУП

Методичні вказівки призначені для надання інформації щодо змісту та порядку виконання лабораторних робіт з дисципліни «Автоматизація оброблення технічної інформації». Цикл лабораторних робіт з даної дисципліни спрямований на отримання практичних навичок роботи з середовищем математичного модулювання MatLAB, вивчення особливостей моделювання цифрових сигналів, дослідження різних методів та підходів до оброблення сигналів.

Отримані під час виконання лабораторних робіт знання та навички дозволять в подальшому проводити автоматизацію оброблення експериментальних даних шляхом створення відповідного програмного забезпечення.

ЛАБОРАТОРНА РОБОТА №1

ЕЛЕМЕНТАРНІ ФУНКЦІЇ ТА ГРАФІЧНІ ПОБУДОВИ

Мета роботи: Ознайомитися з середовищем моделювання MATLAB, особливостями створення та відображення сигналів.

Теоретичні відомості

Короткі теоретичні відомості щодо особливостей роботи в середовищі MatLAB наведено в додатку А.

Також для ознайомлення з програмою можна скористатися наступними посиланнями:

<http://matlab.exponenta.ru/ml/book1/index.php> [1]

<http://matlab.exponenta.ru/index.php>

Початковий код програми з коментарями:

```
% Очищення командного вікна; закриття графічних вікон;  
видалення змінних  
clc; close all; clear;  
  
% Кількість точок розрахунку  
N = 1000;  
% Частота сигналу  
f0 = 1e6;  
% Амплітуда сигналу  
Um = 5.1;  
% Фаза сигналу  
phi = 0;  
  
% Генерування сигналу  
t = (0:pi/N:pi)*1e-6; % Вектор часу в мікросекундах;  
y = Um*sin(2*pi*f0*t+phi); % Досліджуваний сигнал;  
  
% Дискретизація сигналу  
% Для проведення дискретизації сигналу візьмемо кожен 20 точку  
масиву y  
k_d = 20; % коефіцієнт часового прорідження  
yd = y(1:k_d:end);  
  
% Демонстрація квантування дискретного сигналу з кроком  
квантування 1В  
  
yk=fix(yd);  
  
% Виведення графіків на екран  
figure (1) % створення першого вікна для виведення графіку
```

```

subplot(2,1,1) % розбиття вікна на рядки та стовпчики 2x1, та
виведення графіку в першу чарунку

plot (t*1e6, y) %виведення графіку неперервного сигналу, вісь
x в мікросекундах
grid on % включення відображення сітки
title ('Синусоїдальне коливання в часовій області') % назва
графіку
ylabel ('Um, В') % підпис осі y
xlabel ('t, мкс') % підпис осі x
axis ([0 pi 1.1*min(y) 1.1*max(y)]) % розмір області
відображення [xmin xmax ymin ymax]

subplot(2,1,2)
plot (t*1e6, y, '--g') % виведення графіку початкового сигналу
hold on % ввімкнення накладення графіків
stem (t(1:k_d:end)*1e6, yd, '*r') %виведення графіку
дискретного сигналу, вісь x в мікросекундах
stairs (t(1:k_d:end)*1e6, yk) %виведення графіку квантованого
сигналу, вісь x в мікросекундах

legend ({'Початковий сигнал', 'Дискретний сигнал', 'Квантований
сигнал'}, 'location', 'SouthEast')% виведення легенди

grid on
title ('Дискретизований сигнал та квантований сигнал')
ylabel ('Um, В')
xlabel ('t, мкс')
axis ([0 pi 1.1*min(y) 1.1*max(y)])

```

Порядок виконання роботи

1. Створити *m*-файл та скопіювати до нього наведений вище код програми;
2. Використовуючи Інтернет-ресурси або надану літературу, розібратися в призначенні кожної команди програми;
3. Отримати номер індивідуального завдання;
4. Згідно номеру з табл. 1. вибрати необхідні параметри синусоїдальних коливань та згенерувати сигнал, який складається з суми синусоїдальних коливань з заданими параметрами (фазу сигналу обрати довільно в діапазоні від 0 до π).
5. Провести квантування та дискретизацію сигналу.
6. Відобразити на графіках мінімум 2 періоди (умовних періодів, якщо сигнал не періодичний) отриманого складного сигналу.
7. Зробити висновки по роботі.

Зміст звіту

1. Параметри заданих сигналів
2. Лістинг програми
3. Графічні побудови
4. Висновки по роботі

Контрольні питання

1. Поняття аналогового, дискретного та квантованого сигналу.
2. Виконання яких операцій забезпечують команди Вашої програми?

Варіант завдання

Таблиця 1. Параметри синусоїдальних коливань

Номер	Кількість синусоїд	Амплітуда, В	Частота, МГц
1	4	[3.40 0.60 4.30 3]	[4.30 2.10 2.50 0.70]
2	4	[1.30 3.50 2 4.50]	[1 4.90 2.90 3.70]
3	4	[5 1.80 2.40 2.60]	[3.90 4.20 1 1.30]
4	4	[2.10 0.80 2.80 2]	[1.30 1.40 4.60 3.50]
5	4	[2.60 4.60 1 3.90]	[3.80 3 1.30 3.20]
6	3	[1.80 1.10 1.50]	[4.50 0.80 1.60]
7	2	[0.70 2.50]	[0.60 4.50]
8	3	[1.40 0.90 1.90]	[2.60 1 5]
9	2	[2 1.80]	[0.80 1.80]
10	2	[0.70 2.80]	[3.90 3.30]
11	4	[0.90 0.90 4 4.60]	[2.90 1 4.20 2]
12	3	[1.80 3.90 0.50]	[0.70 3.50 3.20]
13	3	[2.90 3.80 3.70]	[4 1.80 3.60]
14	3	[3 2.30 0.80]	[4 2 3.20]
15	3	[3.80 1 1.10]	[3 2.70 4.50]

ЛАБОРАТОРНА РОБОТА №2

РОБОТА З ТАБЛИЧНИМИ ДАНИМИ

Мета роботи: Ознайомитися з способами зчитування табличних даних, їх обробкою та збереженням до файлу.

Теоретичні відомості

Робота з файлами

Найпростішим варіантом зчитування даних, занесених до файлу є використання команди **dlmread**, яка призначена для зчитування числових даних з текстового файлу з ASCII-форматом.

Синтаксис:

`RESULT = dlmread('FILENAME')`

`RESULT = dlmread('FILENAME', 'DELIMITER')`

`RESULT = dlmread('FILENAME', 'DELIMITER', R, C)`

де

`FILENAME` — ім'я файлу;

`DELIMITER` — роздільник між елементами масиву, може бути як звичайні розділові знаки, так і спеціальні символи, наприклад `/t` — горизонтальна табуляція;

`R, C` — починає зчитувати з `R` – рядка, `C` – стовпця. Нумерація починається з 0 елемента.

Для записування даних до файлу можна скористатися командою **dlmwrite**, яка також працює з файлами ASCII-формату.

Синтаксис:

`dlmwrite('FILENAME', M)`

`dlmwrite('FILENAME', M, 'DLM')`

де

`M` — масив даних, які необхідно записати до файлу;

`DLM` — варіант роздільника.

Інтерполяція табличних даних

Інтерполяція — в обчислювальній математиці спосіб знаходження проміжних значень величини за наявним дискретним набором відомих значень.

Для виконання лабораторної роботи у середовищі MatLab рекомендується використовувати наступні функції:

`interp1` — інтерполяція одновимірних табличних даних

Синтаксис:

`yi = interp1 (x, y, xi)`

$y_i = \text{interp1}(x, y, x_i, \text{'METHOD'})$

Опис:

Функція $y_i = \text{interp1}(x, y, x_i)$ будує інтерполюючу криву для одновимірного масиву y , заданого на сітці x ; вихідний масив y_i може бути визначений на більш дрібній сітці x_i . Якщо y — двовимірний масив, то інтерполююча крива будується для кожного рядка окремо. За замовчуванням реалізована лінійна інтерполяція.

METHOD — дозволяє задати метод інтерполяції

'Nearest' — ступінчата

'Linear' — лінійна

'Cubic' — кубічна

'spline' — кубічні сплайни

Прийнято, що аргумент x змінюється монотонно; крім того, для кубічної інтерполяції передбачається, що сітка по x рівномірна.

Порядок виконання роботи

1. Створити m -файл та написати програму, яка буде зчитувати дані з файлів згідно з номером Вашого варіанту N (файли $\text{var}N.\text{txt}$ та $\text{per}N.\text{txt}$), для чого ознайомитися зі структурою файлів та зчитати дані використовуючи функцію `dlmread`;
2. Провести графічне відображення зчитаних даних (файл $\text{var}N.\text{txt}$), переконатися в необхідності проведення інтерполяції табличних даних;
3. Провести ступінчасту, лінійну, кубічну та сплайнову інтерполяцію за допомогою команди `interp1`. В якості нового вектору часу використати діапазон значень: $1e-6:1e-6:1e-3$;
4. Побудувати окремо для кожного методу інтерполяції графіки, які містять точки початкових табличних даних, інтерпольовану криву та тестову криву з файлу $\text{per}N.\text{txt}$ (для тестової кривої вектор часу брати з попереднього пункту), графічна побудова повинна мати вигляд як на рис. 2.1;
5. Знайти абсолютне відхилення інтерпольованої кривої від перевірконої та побудувати гістограму цієї помилки за допомогою функції `hist`;
6. Записати результати інтерполяції до файлу у вигляді таблиці.
7. Зробити висновки по роботі.

Зміст звіту

1. Лістинг програми
2. Графічні побудови
3. Висновки по роботі

Контрольні питання

1. Що таке інтерполяція?
2. Які методи інтерполяції Ви знаєте?
3. Виконання яких операцій забезпечують команди Вашої програми?

Приклад виконання

На рисунку 2.1 наведено приклад графічної побудови для лінійної інтерполяції.

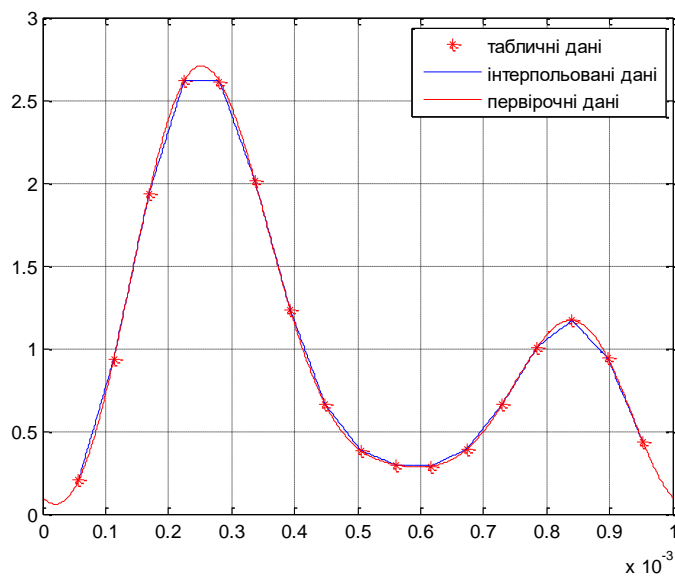


Рисунок 2.1 — Приклад відображення результатів інтеполяції

Подібні графіки повинні бути наведені для кожного методу інтерполяції.

ЛАБОРАТОРНА РОБОТА №3

АПРОКСИМАЦІЯ ТАБЛИЧНИХ ДАНИХ

Мета роботи: ознайомитися з методами апроксимації та дослідити вплив степеня поліному на якість апроксимації. Отримання навичок роботи з Curve Fitting Toolbox

Теоретичні відомості

Апроксимація табличних даних

Для отримання аналітичної залежності, що описує великі масиви даних, використовують методи апроксимації, які ґрунтуються на тому, що масив даних замінюють простою функцією (лінійною, квадратичною, кубічною або іншою), яка не обов'язково проходить через всі експериментальні точки, але описує тенденції зміни цих даних та забезпечує мінімум суми квадратів відхилень експериментальних даних від цієї функції.

Апроксимуючий поліном

В більшості випадків апроксимацію проводять за допомогою степеневого поліному виду:

$$Q(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Підбір коефіцієнтів поліному a_0, a_1, \dots, a_n можна проводити методом найменших квадратів, який полягає у знаходженні такого поліному, щоб сума квадратів відхилення була мінімальною:

$$\sum_{i=0}^N (Q(x_i) - y_i)^2 \Rightarrow \min$$

де y_i — табличні дані; N — кількість табличних даних.

Для виконання лабораторної роботи у середовищі *MATLAB* рекомендується використовувати наступні функції:

polyfit(x, y, n) — повертає вектор коефіцієнтів полінома ***p(x)*** степеня ***n***, який з найменшою середньоквадратичною похибкою апроксимує функцію ***y(x)***. Результатом є вектор-рядок довжиною ***n+1***, що містить коефіцієнти в порядку зменшення степеня. Якщо кількість елементів векторів ***x*** та ***y*** дорівнює ***n+1***, то реалізується звичайна поліноміальна апроксимація, при якій графік полінома точно проходить через вузлові точки з координатами ***(x, y)***, що зберігаються в векторах ***x*** та ***y***. Інакше точний збіг графіка з вузловими точками не спостерігається.

$[p, s]=polyfit(x, y, n)$ — повертає коефіцієнти полінома p та структуру s для використання разом з функцією ***polyval*** для оцінки або передбачення похибки.

polyval(p, x) — повертає значення полінома p в точках, що задані в масиві x . Поліном p – вектор, елементи якого є коефіцієнтами полінома в порядку зменшення степеня x може бути матрицею або вектором. У будь-якому випадку, функція ***polyval*** обчислює значення полінома p для кожного елемента x .

$[y,delta]=polyval(p,x,s)$ – використовує структуру s , повернену функцією ***polyfit*** для оцінки погрішності апроксимації $y\pm delta$.

Створення файлу-функції в *MATLAB*

В середовищі *MATLAB* є можливість створення власних функцій за допомогою файлів-функцій. Файл-функція, це звичайний *m*-файл, який має спеціальний синтаксис та має назву, яка повторює ім'я функції. Файл-функції для успішного використання в інших програмах потрібно зберегти в робочій папці *MATLAB*

Синтаксис файлу-функції:

```
function [y1,...,yN] = myfun(x1,...,xM)  
% Основний коментар (можна переглянути командою lookfor)  
% Додатковий коментар (можна переглянути командою help)
```

Тіло функції з потрібними командами;

```
y1 = вираз;  
...  
yN = вираз;
```

де y_1, \dots, y_N — вихідні параметри функції; x_1, \dots, x_M — вхідні параметри функції.

Всі змінні в файлі-функції мають локальний характер, тому передати дані до функції можна виключно через вхідні параметри, а з функції до основної робочої області — вихідні параметри.

Curve Fitting Toolbox

Curve Fitting Toolbox — пакет розширення *MATLAB* для проведення апроксимації та інтерполяції даних різними методами.

Більш докладно з даним розширенням можна ознайомитися за адресою: <http://matlab.exponenta.ru/curvefitting/> [2]

Порядок виконання роботи

1. Створити *m*-файл та написати програму, яка буде зчитувати дані з файлу згідно з номером Вашого варіанту *V* (файли *varV.txt*) використовуючи команду *dlmread*;

2. Провести графічне відображення зчитаних даних у вигляді окремих точок на графіку, вважаючи, що зчитані дані є значенням певної функції *f(x)*, де $x=1, 2, \dots, N$; *N* — кількість табличних даних

3. Провести апроксимацію степенями поліному 1, 2, 3, 4, 10 за допомогою функцій *polyfit* та *polyval* ;

4. Створити функцію для проведення оцінювання якості апроксимації за допомогою середньоквадратичного відхилення:

$$d = \sqrt{\frac{\sum_{i=0}^N (Q(x_i) - y_i)^2}{N}}$$

Звести результати оцінювання до таблиці та записати її до файлу, з якого буде зрозуміло для якого степеню поліному яке значення відхилення.

Функція повинна бути реалізована у вигляді файлу-функції;

5. Побудувати окремі графіки для кожного з поліномів разом з табличними даними;

6. Запустити графічний інтерфейс розширення Curve Fitting Toolbox за допомогою команди *cftool*. Ознайомитися з інтерфейсом програми.

7. Завантажити до робочої області Ваші табличні дані.

Провести апроксимацію:

- за допомогою поліному (*polynomial*) для різних степеней поліному, порівняти результат з попередньо отриманими даними за допомогою одного з критеріїв придатності наближення (див. «Критерии пригодности приближения» http://matlab.exponenta.ru/curvefitting/3_8.php). Порівняти коефіцієнти поліномів (див. вікно Results).

- провести апроксимацію за допомогою решти параметричних моделей (див. «Параметрические модели» http://matlab.exponenta.ru/curvefitting/3_5.php). Визначивши по графічним побудовам найкращі параметри для кожної моделі апроксимації та виписати в протокол їх критерії придатності наближення.

8. Зробити висновки по роботі.

Зміст звіту

1. Лістинг програми та лістинг створеної функції
2. Графічні побудови

3. Вміст файлу з результатами оцінювання апроксимації
4. Критерії придатності для кожної параметричної моделі
5. Висновки по роботі

Контрольні питання

4. Що таке апроксимація?
5. Якими поліномами крім степеневого може проводитися апроксимація?
6. Виконання яких операцій забезпечують команди Вашої програми?
7. Які основні можливості пакету Curve Fitting Toolbox?

ЛАБОРАТОРНА РОБОТА №4 РОЗРАХУНОК ПАРАМЕТРІВ ФІЛЬТРУ

Мета роботи: ознайомитися з методами побудови фільтрів. Дослідити особливості АЧХ та ФЧХ фільтрів з різними апроксимуючими поліномами.

Теоретичні відомості

Для синтезу цифрового фільтру за аналоговим прототипом можна скористатися наступними функціями: *butter*, *cheby1*, *ellip*, які повертають значення поліномів передавальної характеристики фільтру [3].

Визначення параметрів фільтру

Для визначення мінімального порядку фільтру та частоти зрізу фільтра для отримання потрібної передавальної характеристики існують функції: *buttord*, *cheb1ord*, *ellipord*:

$[n, Wn] = \text{buttord}(Wp, Ws, Rp, Rs)$

$[n, Wn] = \text{buttord}(Wp, Ws, Rp, Rs, 's')$

<http://matlab.exponenta.ru/signalprocess/book1/5/buttord.php>

$[n, Wn] = \text{cheb1ord}(Wp, Ws, Rp, Rs)$

$[n, Wn] = \text{cheb1ord}(Wp, Ws, Rp, Rs, 's')$

<http://matlab.exponenta.ru/signalprocess/book1/5/cheb1ord.php>

$[n, Wn] = \text{ellipord}(Wp, Ws, Rp, Rs)$

$[n, Wn] = \text{ellipord}(Wp, Ws, Rp, Rs, 's')$

<http://matlab.exponenta.ru/signalprocess/book1/5/ellipord.php>

Wp — границя смуги пропускання $Wp=2*fp/Fs$;

Ws — границя смуги загородження $Ws=2*fs/Fs$;

Rp — пульсація АЧХ в смузі пропускання;

Rs — затухання АЧХ в смузі загородження.

Синтез фільтру Батерворта

Синтаксис:

$[b, a] = \text{butter}(n, Wn)$

$[b, a] = \text{butter}(n, Wn, 'ftype')$

$[b, a] = \text{butter}(n, Wn, 'ftype', 's')$

n — порядок фільтру;

Wn — частота зрізу в рад/с для аналогового фільтру та приведена частота, при проектуванні цифрового фільтру (для цифрового фільтру $0 < Wn < 1$);

ftype — параметр, який вказує на тип фільтру. По замовчуванню ФНЧ, також може приймати значення:

high — синтез фільтру високих частот;

stop — синтез режекторного фільтру, порядок фільтру $2 \cdot n$. При цьому параметр Wn повинен бути двохелементним вектором $Wn = [w1 \ w2]$, де $w1$ та $w2$ — верхня та нижня частота смуги загородження.

s — параметр, який вказує, що проектується аналоговий фільтр.

Більш докладно на сайті:

<http://matlab.exponenta.ru/signalprocess/book1/7/butter.php>

Синтез фільтру Чебишева 1 роду

Синтаксис:

$[b, a] = \text{cheby1}(n, Rp, Wn)$

$[b, a] = \text{cheby1}(n, Rp, Wn, 'ftype')$

$[b, a] = \text{cheby1}(n, Rp, Wn, 's')$

$[b, a] = \text{cheby1}(n, Rp, Wn, 'ftype', 's')$

Rp — величина пульсацій в смузі пропускання.

Більш докладно на сайті:

<http://matlab.exponenta.ru/signalprocess/book1/7/cheby1.php>

Синтез еліптичного фільтру

$[b, a] = \text{ellip}(n, Rp, Rs, Wn)$

$[b, a] = \text{ellip}(n, Rp, Rs, Wn, 'ftype')$

$[b, a] = \text{ellip}(n, Rp, Rs, Wn, 's')$

$[b, a] = \text{ellip}(n, Rp, Rs, Wn, 'ftype', 's')$

Rs — величина подавлення в смузі пропускання.

Розрахунок передавальної характеристики

Наведені вище функції повертають коефіцієнти поліномів b та a , які формують передавальну характеристику фільтру. Для отримання передавальної характеристики можна скористатися функцією *freqz* (для цифрового фільтру) та *freqs* (для аналогового):

$h = \text{freqs}(b, a, w)$

$[h, w] = \text{freqs}(b, a)$

$[h, w] = \text{freqs}(b, a, l)$

h — комплексна передавальна характеристика фільтру.

w — вектор частот, для яких проводиться розрахунок в рад/с (1 випадок), або вектор частот для яких функція провела розрахунок (2 та 3 випадок);

l — кількість точок вектора частот.

$[h, w] = \text{freqz}(b, a, l)$

$h = \text{freqz}(b, a, w)$

$[h, f] = \text{freqz}(b, a, l, fs)$

fs — частота дискретизації;

f — вектор частот довжиною f та діапазоном 0 до $fs/2$ Гц.

Побудова АЧХ та ФЧХ фільтру

Для побудови АЧХ (fr) та ФЧХ ($phase$) фільтру за його передавальною характеристикою (h) доречно скористатися наступними функціями:

$fr = 20 * \log_{10}(\text{abs}(h));$

$phase = \text{angle}(h) * 180 / \pi;$

Порядок виконання роботи

1. Отримати номер індивідуального завдання та визначити необхідні параметри для розрахунку фільтрів з таблиці 4.1;
2. Розрахувати мінімальні параметри цифрових або аналогових фільтрів Батерворта, Чебишева та еліптичного фільтрів та занести їх до таблиці.
3. Побудувати АЧХ та ФЧХ розрахованих фільтрів. За допомогою команди `axis` відобразити їх так, щоб виразити особливості кожного з них.
4. Зробити висновки по роботі, звернувши увагу на особливості АЧХ та порядок фільтрів з різним апроксимуючим поліномом.

Зміст звіту

1. Лістинг програми;
2. Таблиця з параметрами фільтрів;
3. Графічні побудови;
4. Висновки по пунктам 2–4.

Контрольні питання

1. Особливості різних типів фільтрів?
2. Параметри фільтрів?
3. Виконання яких операцій забезпечують команди Вашої програми?

Варіант завдання

Таблиця 4.1 Варіанти виконання роботи

№	Тип фільтрів	f_p , кГц	f_s , кГц	R_p , дБ	R_s , дБ	F_s , кГц
1	ФНЧ	5	7	3	50	50
	СФ	[5 6]	[4.5 6.5]	3	20	50
2	ФНЧ	9	11	2	30	50
	РФ	[4.5 6.5]	[5 6]	2	30	50
3	ФВЧ	7	5	2	40	50
	СЧ	[5 7]	[4 8]	2	40	50
4	ФВЧ	20	18	3	60	50
	РФ	[4 8]	[5 7]	3	50	50
5	ФНЧ	12	13	4	80	100
	СФ	[10 15]	[9 16]	2	50	100
6	ФНЧ	9	12	2	50	100
	РФ	[9 16]	[10 15]	1	10	100
7	ФВЧ	40	38	2	40	100
	СЧ	[25 35]	[20 40]	4	50	100
8	ФВЧ	16	15	3	30	100
	РФ	[10 15]	[9 16]	2	30	100
9	ФНЧ	5	7	3	50	50
	СФ	[5 6]	[4.5 6.5]	3	20	50
10	ФНЧ	9	11	2	30	50
	РФ	[4.5 6.5]	[5 6]	2	30	50
11	ФВЧ	7	5	2	40	50
	СФ	[5 7]	[4 8]	2	40	50
12	ФВЧ	20	18	3	60	50
	РФ	[4 8]	[5 7]	3	50	50
13	ФНЧ	12	13	4	80	100
	СФ	[10 15]	[9 16]	2	50	100
14	ФНЧ	9	12	2	50	100
	РФ	[9 16]	[10 15]	1	10	100
15	ФВЧ	40	38	2	40	100
	СФ	[25 35]	[20 40]	4	50	100
16	ФВЧ	16	15	3	30	100
	РФ	[10 15]	[9 16]	2	30	100

ФНЧ — фільтр низьких частот;

ФВЧ — фільтр високих частот;

СФ — смуговий (смугопропускаючий) фільтр;

РФ — режекторний (смугозагороджуючий) фільтр;

ЛАБОРАТОРНА РОБОТА №5

ФІЛЬТРАЦІЯ СИГНАЛІВ

Мета роботи: Побудувати цифровий фільтр та дослідити результати фільтрації в часовій та частотній областях.

Теоретичні відомості

Цифрова фільтрація сигналу

Для фільтрування сигналу за допомогою цифрового фільтру в MATLAB використовується функція *filter*

$$y = \text{filter}(b, a, x)$$

x — сигнал на вході фільтру;

y — сигнал на виході;

a, b — коефіцієнти розрахованого фільтру.

Спектр сигналу

Для дослідження спектру відфільтрованого сигналу використати алгоритм дискретного перетворення Фур'є, для цього можна скористатися наступним кодом:

```
%Спектральний аналіз
f = Fs*(0:N-1)/(2*N-1); % Вектор частот для графіку, Гц
% Розрахунок спектру модулюючого сигналу швидким перетворенням
Фур'є.
sp_1 = abs(fft(S)); % Модуль швидкого перетворення Фур'є, для
визначення спектральних амплітуд
sp_1 = sp_1/N; % Нормування амплітуд спектру
sp_1(1) = sp_1(1)/2; % Нормування постійної складової
stem(f, sp_1(1:N)) % Виведення спектральних складових на екран
```

Варто врахувати, що кількість точок для розрахунку повинна бути в два рази більша кількості відліків перетворення Фур'є (N) та мати значення рівне степені двійки.

Порядок виконання роботи

1. Отримати номер індивідуального завдання;
2. Згенерувати сигнал, як суму синусоїдальних коливань з параметрами згідно Вашого варіанту, табл. 5.1. Задати наступні параметри: $N = 512$ — кількість відліків перетворення Фур'є; $t = \text{linspace}(0, ts, 2*N)$ — вектор часу від 0 до ts (тривалість 5–10 періодів згенерованого сигналу). Зверніть увагу, згенерований сигнал повинен містити повний період (періоди) за розрахований Вами час ts .
3. Створити цифрові ФНЧ з заданими параметрами (f_c — частота зрізу, порядок фільтра $n = 1, 2, 5$, частоту дискретизації F_s визначити, як

обернене значення до інтервалу дискретизації згенерованого вектору часу) та пропустити через нього згенерований сигнал.

4. Побудувати на одному графіку початковий сигнал, сигнал після фільтру та ідеально відфільтрований сигнал (сума тих синусоїдальних складових, які б залишилися при проходженні через ідеальний фільтр). Повторити для кожного порядку фільтру. Межі для кожного графіку обрати так, щоб було видно 1–2 періоди сигналу

5. Побудувати АЧХ фільтру з різним значенням n .

6. Розрахувати спектр початкового сигналу та спектри відфільтрованих сигналів. Відобразити їх на окремих графіках, промасштабувавши осі так, щоб відобразити корисні складові спектру.

7. Повторити пункти 3 та 6 для ФВЧ.

8. Зробити висновки по роботі.

Контрольні питання

1. Вплив порядку фільтру на відфільтрований сигнал
2. Що таке спектр сигналу?
3. Особливості представлення сигналів в часовій та частотній області?
4. Особливості алгоритму швидкого перетворення Фур'є
5. Чому спектр відфільтрованого сигналу містить «зайві» частотні складові?

Варіант завдання

Таблиця 5.1 Параметри сигналу та фільтру

№	U_m	F , Гц	f_c , Гц	Тип полін.
1	[1 0.5 0.25 0.1]	[20 30 60 100]	50	Чеб
2	[1 0.5 0.25 0.1]	[20 30 60 100]	50	Бат
3	[1 1 0.5 0.25]	[10 20 40 80]	30	Чеб
4	[1 1 0.5 0.25]	[10 20 40 80]	30	Бат
5	[1 0.75 0.5 0.25]	[20 40 60 80]	50	Чеб
6	[1 0.75 0.5 0.25]	[20 40 60 80]	50	Бат
7	[1 0.5 0.2 0.1]	[10 20 40 100]	30	Чеб
8	[1 0.5 0.2 0.1]	[10 20 40 100]	30	Бат
9	[1 0.5 0.25 0.1]	[20 30 60 100]	50	Чеб
10	[1 0.5 0.25 0.1]	[20 30 60 100]	50	Бат
11	[1 1 0.5 0.25]	[10 20 40 80]	30	Чеб
12	[1 1 0.5 0.25]	[10 20 40 80]	30	Бат
13	[1 0.75 0.5 0.25]	[20 40 60 80]	50	Чеб
14	[1 0.75 0.5 0.25]	[20 40 60 80]	50	Бат
15	[1 0.5 0.2 0.1]	[10 20 40 100]	30	Чеб
16	[1 0.5 0.2 0.1]	[10 20 40 100]	30	Бат

ЛАБОРАТОРНА РОБОТА №6

ОБРОБКА ЗОБРАЖЕННЯ

Мета роботи: отримання практичних навичок роботи з цифровими зображеннями та їх оброблення для подальшого використання.

Теоретичні відомості

Зображення являється двовимірною функцією $f(x, y)$, де x та y — це просторові координати, а значення f — інтенсивність (яскравість) чи рівень сірого (для монохромних зображень) кольору зображення в точці з координатами x та y . Якщо змінні x , y та f приймають дискретні значення, то дане зображення є цифровим.

Під цифровою обробкою зображення розуміють обробку цифрових зображень за допомогою цифрових обчислювальних машин. Цифрове зображення складається з кінцевої кількості елементів, кожен з яких розміщений у визначеному місці та має певне значення. Традиційно такі елементи називаються пікселями (*pixel*, скорочено від англ. *PICTure'S ELe ment* — елемент зображення).

В цифровій обробці зображень розрізняють три рівні процесу обробки: низький, середній та високий.

Низький рівень обробки зображення — це такий рівень, при якому на вході та виході процесу обробки є зображення, а в процесі обробки відбуваються елементарні операції: зниження шуму, підвищення контрастності чи збільшення чіткості.

Середній рівень — це сегментація (розділення зображення на окремі частини), опис, стиснення, класифікація зображення та ін. операції. Якщо на вході процесу обробки є зображення, то на виході — атрибути та ознаки зображення, отримані в процесі обробки (контури, границі, класифікаційні ознаки та ін.).

Високий рівень — це рівень осмислення розпізнаних (класифікованих) об'єктів з переходом до інших когнітивних функцій.

Так при розпізнаванні сканованого тексту одразу проводиться перший рівень: підвищення чіткості, зниження зашумленості, потім відбувається перехід до другого рівня — виокремлення області з текстом, розбиття її на окремі рядки та літери, з подальшим їх розпізнаванням. Високим рівнем є осмислення тексту, проте даний процес більш-менш вдається тільки людині.

Робота з зображеннями в *MATLAB*

В середовищі *MATLAB* прийнято розбивати зображення на рядки ($r \rightarrow 1 \dots M$) та стовпчики ($c \rightarrow 1 \dots N$) по пікселям, починаючи з верхнього лівого кута (див. рис. 6.1) [4]

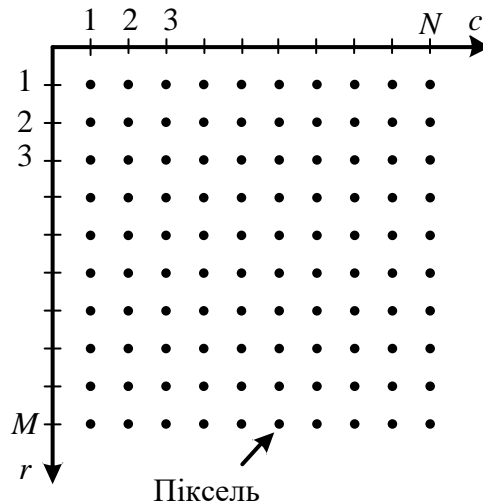


Рисунок 6.1 — Особливості розбиття зображення на пікселі

Тоді будь яке зображення, розміром $N \times M$ можна представити матрицею:

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \dots & f(M,N) \end{bmatrix}$$

де $f(r,c)$ — рівень сірого для пікселя з координатами $r \times c$.

При оперуванні з кольоровими зображеннями в RGB-представленні (повнокольорове зображення) матриця стає тривимірною $[M, N, 3]$, де $[M, N, 1]$ — яскравість червоного, $[M, N, 2]$ — зеленого, $[M, N, 3]$ — синього кольору. Більш докладно про способи представлення зображення можна прочитати за адресою:

http://matlab.exponenta.ru/imageprocess/book5/6_1.php

Значення яскравості може бути задано одним з форматів (класів): uint8, uint16, int16, single та double.

Ілюстрація формування кольорового зображення

Виконайте нижче наведений код, який ілюструє одну з повнокольорових палітр зображення.

```
RGB=reshape(ones(128,1)*reshape(jet(128),1,384),[128,128,3]);  
R=RGB(:,:,1);
```

```

G=RGB(:,:,2);
B=RGB(:,:,3);
figure(1)
subplot (221)
imshow(RGB)
title('Повноколірне зображення')
subplot (222)
imshow(R)
title('Червона складова')
subplot (223)
imshow(G)
title('Зелена складова')
subplot (224)
imshow(B)
title('Синя складова')

```

Завантаження зображення

Завантаження зображення з файлу виконується за допомогою команди `imread`, яка має синтаксис [5]:

```
img = imread('filename', 'fmt')
```

де *filename* — ім'я файлу зображення (допускається з вказуванням шляху, наприклад: 'E:\tmp\img.jpg')

fmt — розширення файлу (не обов'язково).

<http://matlab.exponenta.ru/imageprocess/book3/3/imread.php>

Визначення параметрів зображення

Для визначення розміру матриці зображення можна скористатися функцією `size`, синтаксис:

```
size(img)
```

```
[M, N]=size (img)
```

де *img* — змінна, в якій зберігається матриця зображення

Щоб отримати розширену інформацію про матрицю зображення використовують функцію `whos`, в якості аргументу функції є матриця зображення.

Для читання з файлу інформації про зображення використовується функція `imfinfo`, синтаксис подібний до функції `imread`

<http://matlab.exponenta.ru/imageprocess/book3/3/imfinfo.php>

Виведення зображення на екран

Виведення зображення на екран виконується за допомогою функції `imshow`, синтаксис:

```
imshow(img, n)
```

де *img* — матриця зображення,

n — кількість відтінків сірого для монохромного зображення (по замовчуванню 256), не обов'язковий параметр. Для виведення на екран кольорового зображення даний параметр не вказується.

<http://matlab.exponenta.ru/imageprocess/book3/5/imshow.php>

Запис зображення до файлу

Для збереження результатів перетворення використовується функція *imwrite*, яка має синтаксис:

imwrite(img, filename, fmt)

<http://matlab.exponenta.ru/imageprocess/book3/3/imwrite.php>

Гістограма розподілу інтенсивності пікселів

Гістограма розподілу інтенсивності пікселів зображення є одним з головних показників, який дозволяє проаналізувати якість зображення з точки зору контрастності. Вважається, що якісне зображення має рівномірну гістограму розподілу інтенсивностей. Для отримання гістограми використовується функція *imhist(img, n)*, де n — кількість стовпчиків гістограми (по замовчуванню 256)

<http://matlab.exponenta.ru/imageprocess/book3/9/imhist.php>

Для коригування гістограми розподілу (і всього зображення в цілому) використовуються функції:

histeq — еквалізація (лінеризація) гістограми:

<http://matlab.exponenta.ru/imageprocess/book3/10/histeq.php>

adapthisteq — адаптивне вирівнювання гістограми:

<http://matlab.exponenta.ru/imageprocess/book3/10/adapthisteq.php>

imadjust — перетворення діапазону яскравостей елементів зображення:

<http://matlab.exponenta.ru/imageprocess/book3/10/imadjust.php>

http://matlab.exponenta.ru/imageprocess/book5/5_6.php

Особливості використання кожного з перерахованих методів наведено на сайті:

<http://matlab.exponenta.ru/imageprocess/book2/29.php>

Повний перелік функцій для роботи з зображеннями в *MATLAB* можна переглянути за адресою:

<http://matlab.exponenta.ru/imageprocess/book3/index.php>

Порядок виконання роботи

1. Завантажити кольорове зображення до середовища *MATLAB*, переглянути отриману матрицю зображення (Ім'я файлу формується як: група латиницею (*RV*, *RB*, *RI*) та номер за списком. Файли знаходяться відповідно у теках *RV*, *RB*, *RI*);

2. Переглянути інформацію про файл (*imfinfo*) та матрицю зображення (*whos*), пояснити чому відрізняється розмір в байтах;

3. За допомогою функції *rgb2gray* перетворити кольорове зображення на монохромне. Порівняти матриці кольорового та монохромного зображення. З'ясувати яке числове значення відповідає білому, а яке чорному кольору;

4. Завантажити спотворене зображення згідно з варіантом (див. п.1, до сформованого імені додається літера «р»). Переглянути гістограми інтенсивностей спотвореного та отриманого в п. 3 зображення;

5. Використовуючи команди *histeq* та *imadjust*, отримати зі спотвореного зображення таке зображення, яке візуально буде максимально схожим на зображення з п. 3. (Параметри даних функцій підібрати самостійно), переглянути їх гістограми;

6. Зберегти отримане зображення з п. 5 на диск. Зчитати інформацію про файл та порівняти її з інформацією файлу зображення з п. 4.

7. Для магістрів! Відкрити зображення графіку функції (файл, який складається з «рег» та номеру по списку. Знаходиться в теці «magistr») та розробити алгоритм «оцифровування» даного графіку. Тобто отримати з картинки набір табличних даних, за якими можна побудувати аналогічний графік. Написати програму, яка реалізує даний алгоритм (за працюючу програму додаткові бали).

Зміст звіту

1. Лістинг програми;
2. Інформація про файли з п.2;
3. Графічні побудови:
 - в одному графічному вікні розмістити монохромне зображення з п.3 та п.4 та відповідні їм гістограми;
 - в одному графічному вікні (або в двох) розмістити монохромні зображення з п.3 та п.5 (отримані за допомогою обох функцій) та відповідні їм гістограми;
4. Магістри наводять запропонований алгоритм та програму;
5. Зробити висновки по п. 2, 3, 6.

Контрольні питання

1. Виконання яких операцій забезпечують команди Вашої програми?
2. Чому розмір файлу зображення збереженого на диску відрізняється від розміру відкритого зображення?
3. Як відбувається перетворення кольорового зображення в монохромне?

ЛАБОРАТОРНА РОБОТА №7

МОДУЛЯЦІЯ СИГНАЛІВ

Мета роботи — дослідження амплітудної та частотної модуляції та квадратурної маніпуляції.

Короткі теоретичні відомості

Канал зв'язку характеризується ефективним діапазоном частот, що визначає той частотний діапазон сигналу, який може передаватися за допомогою даного каналу зв'язку. Часто корисний сигнал лежить за межами цього діапазону (в більшості випадків нижче потрібного діапазону), тому для ефективного передавання використовують модуляцію сигналу. Модуляція сигналу — це процес зміни в часі за заданим законом параметрів якогось з фізичних процесів. В радіотехніці таким фізичним процесом є гармонійне коливання, яке називають частотою носієм (несною частотою, опорною частотою):

$$s(t) = A_0 \cos(\omega_0 t + \varphi_0)$$

де A_0 — амплітуда, ω_0 — частота, φ_0 — початкова фаза, t — час.

Зміна одного з вищеперерахованих параметрів визначає тип модуляції: амплітудну, частотну чи фазову. Часто частотну та фазову модуляцію поєднують під однією назвою — кутова модуляція.

При амплітудній модуляції змінюється амплітуда частоти носія у відповідності до закону зміни модулювального сигналу $S_M(t)$, тобто $A(t) = kS_M(t)$, де k — коефіцієнт пропорційності, тоді АМ коливання визначається за формулою:

$$S_{AM}(t) = A(t) \cdot \cos(\omega_0 t + \varphi_0)$$

Якщо модулювальний сигнал приймає від'ємні значення, то до нього додають постійну складову, щоб уникнути явища перемодуляції:

$$A(t) = A_0 + kS_M(t)$$

У випадку, коли модулювальним сигналом є гармонійне коливання $S_M(t) = A_s \cos(\omega_s t + \varphi_s)$ обвідна визначається за формулою:

$$A(t) = A_0 + \Delta A_m \cos(\omega_s t + \varphi_s)$$

де $\Delta A_m = k \cdot A_s$ — амплітуда зміни обвідної.

Відношення $M = \frac{\Delta A_m}{A_0}$ називається коефіцієнтом модуляції. В

загальному випадку коефіцієнт модуляції визначається за формулою:

$$M = \frac{A_{\max} - A_{\min}}{A_{\max} + A_{\min}}$$

де A_{\max}, A_{\min} — відповідно максимальна і мінімальна амплітуда обвідної.

Таким чином, для АМ сигналу з гармонійним модулювальним сигналом можемо записати вираз:

$$S_{AM}(t) = A_0(1 + M \cdot \cos(\omega_s t + \varphi_s)) \cos(\omega_0 t + \varphi_0) \quad (1)$$

При кутовій модуляції у відповідності з модулювальним сигналом $S_M(t)$ змінюється частота (ЧМ, англійською FM) або початкова фаза (ФМ, англійською PM) опорного сигналу, так при фазовій модуляції:

$$\varphi(t) = kS_M(t)$$

Тоді ФМ-сигнал визначається за формулою:

$$S_{PM}(t) = A \cos(\omega_0 + kS_M(t)) \quad (2)$$

Весь аргумент функції \cos називається повною фазою коливання:

$$\Psi(t) = \omega_0 + kS_M(t)$$

Миттєва частота сигналу з кутовою модуляцією визначається за формулою:

$$\omega(t) = \frac{d\Psi}{dt}$$

Тобто у випадку фазової модуляції змінюється не лише початкова фаза коливання, а ще й частота. Повну фазу коливання можна визначити за формулою:

$$\Psi(t) = \int \omega(t') dt'$$

При частотній модуляції модулювальний сигнал лінійно пов'язаний з миттєвою частотою коливання:

$$\omega(t) = \omega_0 + kS_M(t)$$

Тоді повна фаза коливання визначається шляхом інтегрування:

$$\Psi(t) = \omega_0 t + k \int S_M(t') dt' + \varphi_0$$

Відповідно ЧМ-сигнал розраховується за формулою:

$$S_{FM}(t) = A \cos(\omega_0 t + k \int S_M(t') dt' + \varphi_0) \quad (3)$$

У випадку гармонійного модулювального сигналу початкова фаза змінюється по гармонійному закону:

$$\varphi(t) = \beta \sin(\omega_s t)$$

Коефіцієнт β називається індексом кутової модуляції. Він визначає інтенсивність коливання початкової фази. Тоді миттєва частота визначається як:

$$\omega(t) = \frac{d\Psi(t)}{dt} = \omega_0 + \beta\omega_s \cos(\omega_s t)$$

З даної формули випливає, що максимальне відхилення миттєвої частоти від середнього значення ω_0 складає $\beta\omega_s$. Ця величина називається девіацією частоти та позначається ω_d .

Відповідно індекс кутової модуляції визначається за формулою:

$$\beta = \frac{\omega_d}{\omega_s}$$

MatLAB містить декілька вбудованих функцій для генерування модульованих сигналів. Універсальною є функція `modulate`:

`y = modulate(x, fc, Fs, 'method', opt)`

де: x — модулювальний сигнал;

fc — частота опорного сигналу;

Fs — частота дискретизації;

`opt` — додатковий параметр, в залежності від методу;

'method' — цей параметр може приймати такі значення:

`amdsb-sc` чи `am` — амплітудна модуляція з придушенням частоти-носія;

`amdsb-tc` — АМ з частковим придушенням частоти-носія, параметр `opt` задає ступінь придушення;

`amssb` — АМ модуляція з однією бічною частотою;

`fm` — частотна модуляція, параметр `opt` — коефіцієнт k з формули (3), якщо параметр не вказаний, то він визначається за формулою: $opt = (fc / Fs) \cdot 2\pi / \max(abs(x))$. Максимальна девіація частоти не перевищує fc ;

`pm` — фазова модуляція, параметр `opt` — коефіцієнт k з формули (2), якщо параметр не вказаний, то він визначається за формулою: $opt = \pi / \max(abs(x))$, максимальне відхилення за фазою не перевищує π радіан;

`pwm` — широтно-імпульсна модуляція;

`ptm` — фазо-імпульсна модуляція;

`qam` — квадратурна модуляція.

Проте використання даної функції менш наочне з точки зору зміни основних параметрів модульованих коливань, таких як девіація частоти та фази, тому доцільно використовувати функцію `fmmod` для частотної модуляції, а функцію `pmmmod` — для фазової.

$y = \text{fmmod}(x, fc, Fs, \text{freqdev})$

де: x — модулювальний сигнал;

fc — частота опорного сигналу;

Fs — частота дискретизації;

freqdev — девіація частоти в Гц;

$y = \text{pmmmod}(x, fc, Fs, \text{phasedev})$

де: phasedev — девіація фази в радіанах;

Порядок виконання роботи

1. Згенерувати модулювальний сигнал як суму косинусоїдальних складових згідно з варіантом завдання, див. табл. 7.1. Задати наступні параметри: $N = 512$ — кількість відліків перетворення Фур'є; $t = \text{linspace}(0, ts, 2*N)$ — вектор часу від 0 до ts (тривалість двох-трьох періодів згенерованого сигналу). Зверніть увагу, згенерований сигнал повинен містити повний період (періоди) за розрахований Вами час ts .

2. Визначити частоту дискретизації Fs як обернене значення до інтервалу дискретизації згенерованого вектору часу.

3. За формулою (1) створити амплітудно-модульований сигнал, з коефіцієнтом модуляції 0,5. Побудувати модулювальний сигнал та АМ-сигнал з обвідною в одній фігурі, як показано на рис. 7.1. Використовуючи швидке перетворення Фур'є, розрахувати спектр даного модульовального сигналу та АМ-сигналу. Побудувати розраховані спектри та за допомогою команди *axis* виділити зі спектру інформативну складову (рис. 7.2).

4. Створити АМ-сигнал з коефіцієнтом модуляції 1. Побудувати на одній фігурі цей сигнал в часовій та частотній області (рис. 7.3). Відмітити основні відмінності з попереднім сигналом.

5. Дослідити вплив постійної складової модульовального сигналу на його спектр та спектр АМ-сигналу. Дослідити, що відбудеться зі спектром сигналу, якщо за заданий час ts модулювальний сигнал міститиме не цілий період (періоди). Пояснити результати.

6. За допомогою команди *fmmod* згенерувати частотно-модульований сигнал, розрахувати його спектр. Дослідити вплив значення девіації частоти на вигляд сигналу в часовій та частотній області. Побудувати графік ЧМ-сигналу в часовій та частотній області (рис. 7.3) при такому

значенні девіації частоти, щоб частотних складових було мало, а в часовій області можна було спостерігати зміну частоти опорного коливання.

7. Дослідити вплив амплітуди модулювального сигналу на ЧМ-сигнал в часовій та частотній області. Пояснити результати.

8. За допомогою команди *rttmod* згенерувати фазо-модульований сигнал та розрахувати його спектр. Дослідити вплив девіації фази на вигляд ФМ-сигналу в часовій та частотній області. Побудувати графік ФМ-сигналу в часовій та частотній області (рис. 7.3) при значенні девіації фази $\pi/2$.

9. Зробити висновки по роботі (п. 4-8).

Зміст звіту

1. Лістинг програми;
2. Всі необхідні графічні побудови;
3. Висновки по роботі.

Контрольні питання

1. АМ-сигнал, метод отримання, ширина спектру?
2. Перемодульований АМ-сигнал? Його вигляд в часовій та частотній області?
2. На що впливає девіація частоти при ЧМ-модуляції?
3. На що впливає амплітуда модулювального сигналу при ЧМ-модуляції?
4. Що таке ФМ-модуляція?

Варіант завдання та приклад виконання

Таблиця 7.1 Параметри сигналів згідно з варіантом

Варіант	Амплітуда модулювального сигналу	Частота модулювального сигналу, Гц	Частота опорного сигналу f_c , Гц
1	[0.75 0.5]	[500 750]	2000
2	[0.75 0.5]	[400 600]	2000
3	[0.75 0.5]	[200 300]	1000
4	[0.5 0.75]	[200 300]	1000
5	[0.8 0.6]	[250 500]	1000
6	[0.9 0.5]	[250 500]	1000
7	[0.9 0.5]	[500 750]	1000
8	[0.5 0.75]	[500 750]	2000
9	[0.5 0.75]	[400 800]	2000
10	[0.5 0.75]	[500 1000]	2000

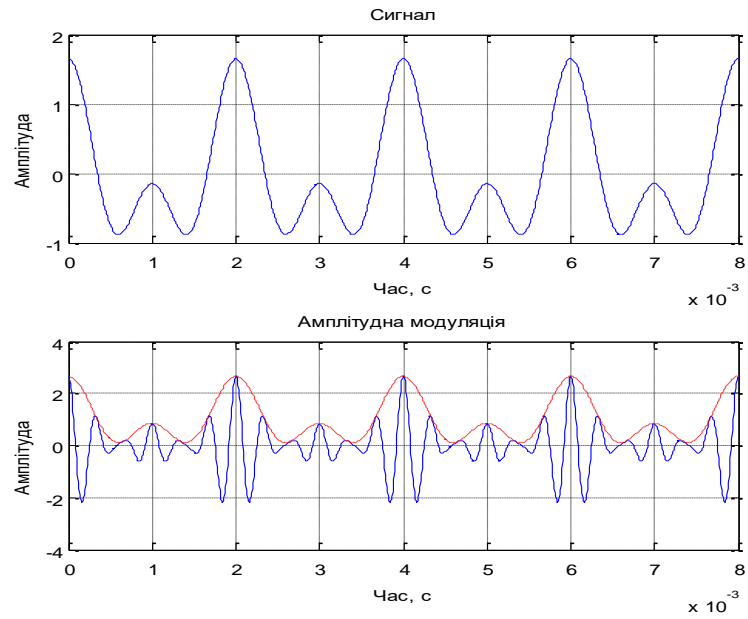


Рисунок 7.1 — Модульовальний та АМ сигнали.

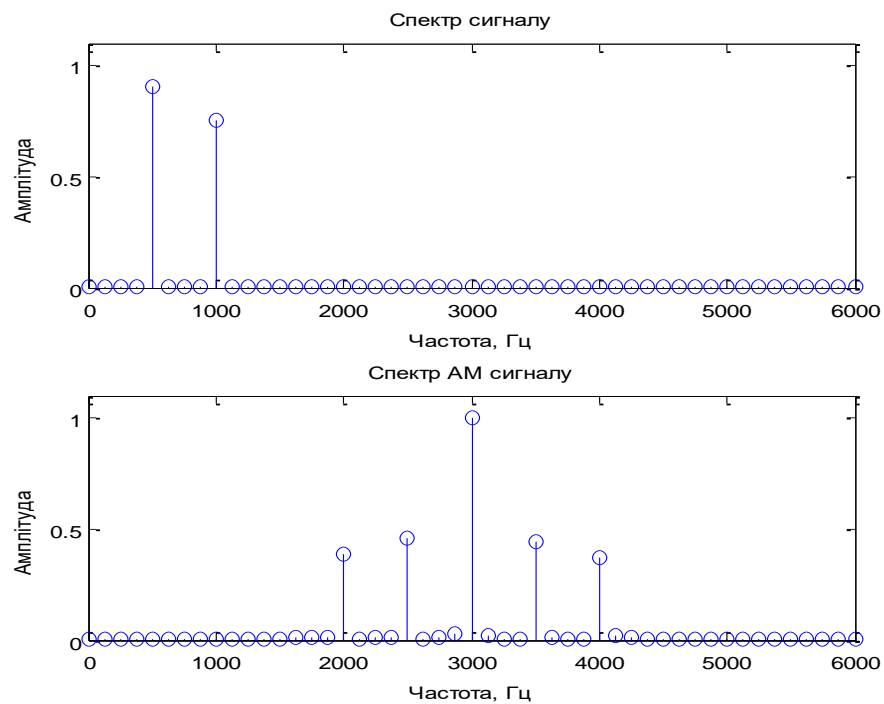


Рисунок 7.2 — Спектр модульовального та АМ сигналу

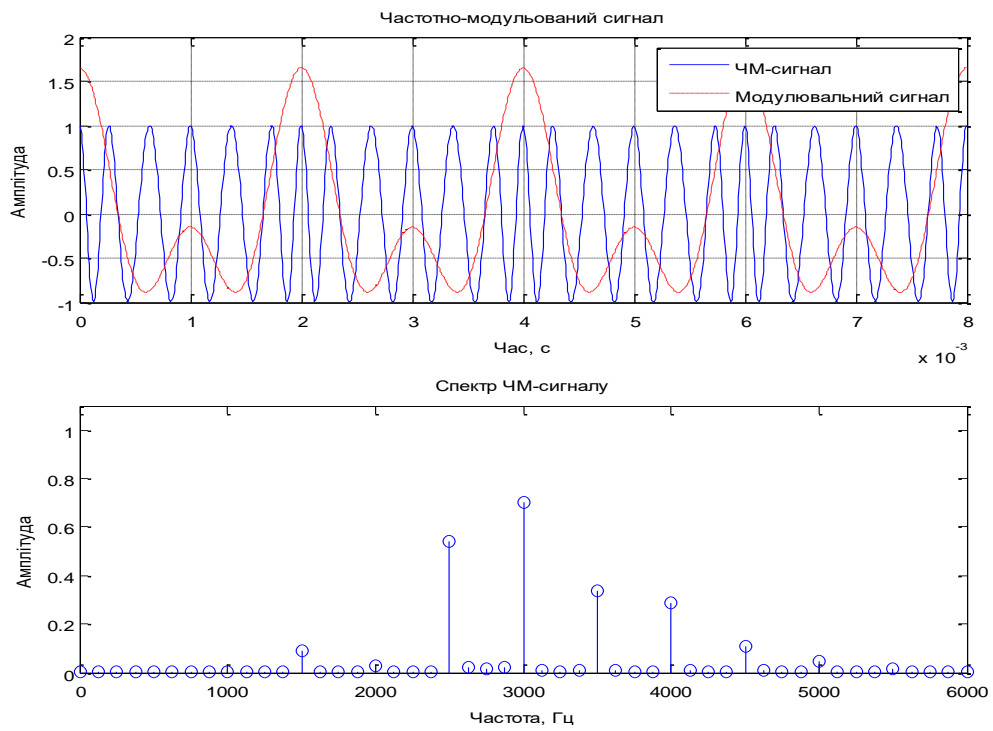


Рисунок 7.3 — ЧМ-сигнал та його спектр

ПЕРЕЛІК ПОСИЛАНЬ

1. Потемкин В. Г. Введение в Matlab (v 5.3) / В. Г. Потемкин [Електронний ресурс] — Режим доступу: <http://matlab.exponenta.ru/ml/book1/index.php> — Назва з екрану.
2. Ануфриев И. Е. Основы работы в Curve Fitting Toolbox / И. Е. Ануфриев [Електронний ресурс] — Режим доступу: <http://matlab.exponenta.ru/curvefitting/>.
3. Список функций Signal Processing Toolbox — Режим доступу: <http://matlab.exponenta.ru/signalprocess/book1/index.php> — Назва з екрану.
4. Описание Image Processing Toolbox — Режим доступу: <http://matlab.exponenta.ru/imageprocess/book5/index.php> — Назва з екрану.
5. Список функций Image Processing Toolbox — Режим доступу: <http://matlab.exponenta.ru/imageprocess/book3/index.php> — Назва з екрану

ДОДАТОК А. ОСОБЛИВОСТІ РОБОТИ В MATLAB

MatLAB (Matrix Laboratory) — високопродуктивна мова для технічних розрахунків. Вона включає в себе розрахунки, візуалізацію та програмування у зручному середовищі, де задачі та рішення виражаються у формі, близькій до математичної (рис. А.1).

MatLAB — це інтерактивна система, в якій основним елементом даних є масив. Це дозволяє вирішувати різноманітні задачі, пов'язані з технічними розрахунками, особливо з тими, де використовуються матриці та вектори, в декілька разів швидше, ніж при використанні скалярних мов програмування.

Типове використання MatLAB — математичні розрахунки, створення алгоритмів, моделювання, аналіз та візуалізація даних, наукова та інженерна графіка, розробка прикладних програм.

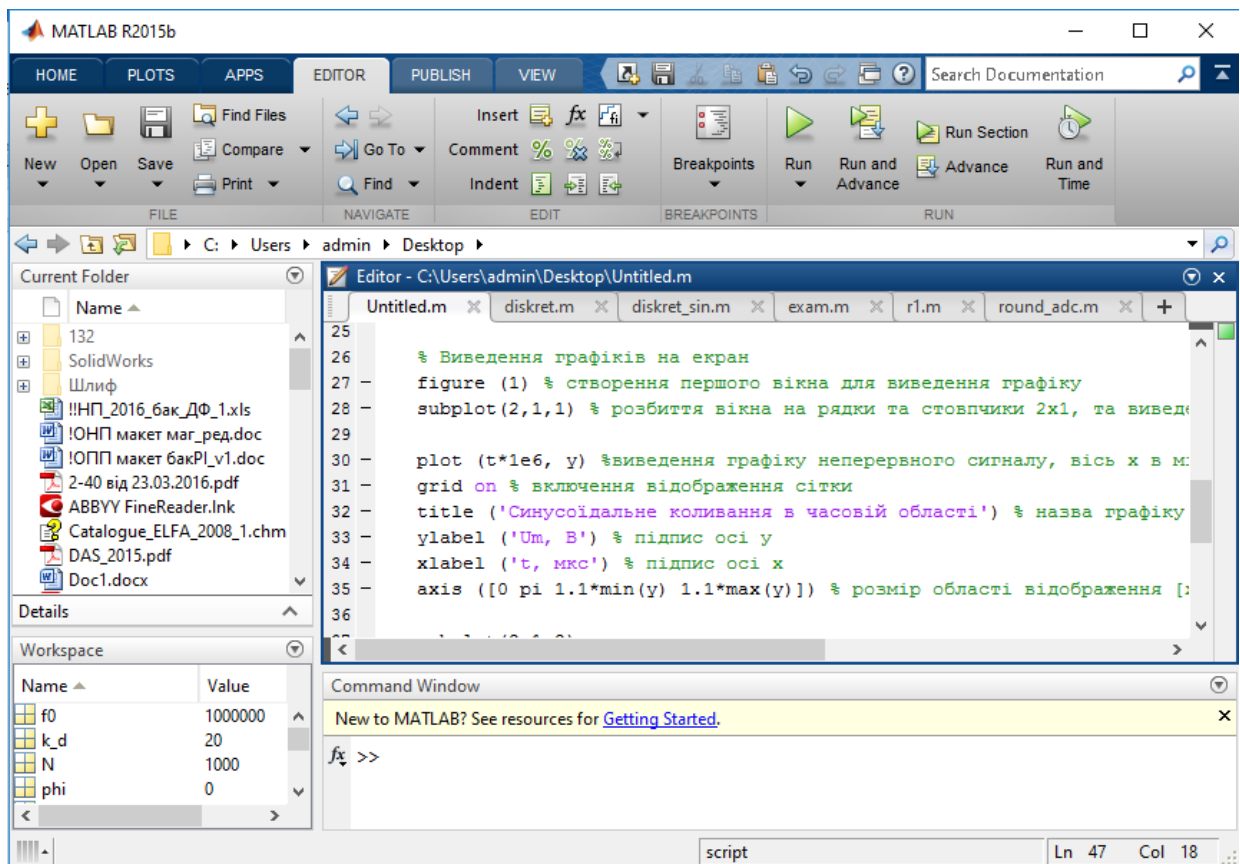


Рисунок А.1 — Зовнішній вигляд вікна програми

Система MatLAB складається з п'яти основних частин:

1. Мова MatLAB — високорівнева мова матриць та масивів.
2. Середовище MatLAB — набір інструментів та засобів для створення та налаштування програм.

3. Графіка — для дво- та тривимірного відображення даних, а також для візуалізації інтерфейсів створюваних програм.
4. Бібліотека математичних функцій — від елементарних до складних функцій.
5. Програмний інтерфейс — бібліотеки, які дозволяють писати програми на СІ та Фортран.

Також в MatLAB присутня програма Simulink — інтерактивна система для моделювання нелінійних динамічних середовищ.

Формати чисел

Зміна формату чисел проводиться за допомогою відповідної команди (табл. А.1), виконаної в командному рядку, впливає виключно на відображення результату. Розрахунки завжди проводяться з подвійною точністю! Решту форматів можна переглянути за допомогою команди `help format`.

Таблиця А.1 Деякі формати чисел MatLAB

Назва	Команда	Вигляд
Короткий формат	<code>format short</code>	0.3333
Коротка експоненціальна	<code>format short e</code>	3.3333e-001
Довгий формат	<code>format long</code>	0.3333333333333333
Довга експоненціальна	<code>format long e</code>	3.333333333333333e-001
Банківський формат	<code>format bank</code>	2.00

Символи та оператори MATLAB

Спеціальні символи:

`:` — оператор створення масиву даних, наприклад, `x:y` — аналогічно вектору $[x, x+1, x+2, \dots, y]$, `x:k:y` — $[x, x+k, x+2k, \dots, y]$;

`()` — використовується для задавання аргументу функції, порядку виконання операцій в математичних виразах, вказування індексів масивів;

`[]` — використовується для формування векторів та матриць;

`{}` — формування масивів чарунок;

`.` — слугує для відокремлення цілої частини від дробової та для виконання операцій по елементно між масивами;

`;` — використовується в кінці операторів для запобігання виведення результату на екран, а також для відокремлення рядків матриці;

`,` — для розділення елементів матриці та вектора;

`%` — початок коментарів;

`=` — символ рівності для присвоювання значення змінній;

`'` — для відокремлення символьних змінних;

! – вказівник на те, що наступною буде команда операційної системи;
 .. – перехід на один рівень вверх по дереву каталогів;
 ... – продовження строки, використовується для перенесення тексту на наступний рядок.

Оператори відношення:

Використовуються для порівняння двох операндів. Значення 1/0 (вірно/не вірно, true/false).

Функція	Ім'я оператора	Позначення
eq	Рівно	==
ne	Не рівно	~=
lt	Менше	<
gt	Більше	>
le	Менше або рівно	<=
ge	Більше або рівно	>=

Логічні оператори:

Функція	Ім'я	Позначення
and	І	&
or	або	
not	ні	~
xor	або-ні	-

Змінні в MATLAB

Змінні — це символи, що використовуються для позначення даних, які необхідно зберегти. Ім'я змінної (ідентифікатор) повинне бути не більше за 31 символ (літери, цифри та деякі дозволені символи) і обов'язково починатися з літери. Всі змінні MatLAB мають матричний вигляд (в найпростішому випадку це матриця 1×1). Переглянути змінні можна у вікні робочої області (WorkSpace) MatLAB.

Для очищення змінних можна використовувати команду *clear*:

clear – знищення всіх змінних;

clear x – знищення змінної *x*;

clear x y z – знищення групи змінних.

Функції

MatLAB містить велику кількість вбудованих функцій та дозволяє програмісту розроблювати свої функції. Список елементарних функцій можна отримати за допомогою команди *help elfun*.

До елементарних функцій MatLAB відносяться:

Тригонометричні функції;

Експоненціальні, логарифмічні та степеневі функції;
Функції для роботи з комплексними числами;
Функції округлення та визначення залишку від ділення.

Робота з матрицями

Створення та робота з матрицями:

$A = [x_{11} \ x_{12} \ x_{13}; x_{21} \ x_{22} \ x_{23}; x_{31} \ x_{32} \ x_{33}]$

eye(n), *eye(n, m)* – створення одиничної матриці;

ones(n, m) – матриця зі всіма одиницями;

zeros(n, m) – матриця зі всіма нулями;

magic(n) – магічний квадрат $n \times n$.

linspace(x, y) – масив зі 100 точок від x до y ;

linspace(x, y, k) – масив з k точок від x до y .

logspace(a, b) – масив з 50 точок логарифмічному масштабі між декадами 10^a та 10^b ;

logspace(a, b, n) – масив з n точок логарифмічному масштабі декадами 10^a та 10^b .

randperm(n) – масив з випадкових перестановок цілих чисел від 0 до n ;

rand(n) – матриця $n \times n$ випадкових чисел від 0 до 1 (*rand(m, n, p...)*);

sum(A) – повертає суму елементів масиву, якщо A — вектор, або вектор-рядок, що містить суму кожного стовпчика, якщо A — матриця;

sum(A, dim) – повертає суму елементів по стовпчикам ($dim = 1$), рядкам ($dim = 2$).

det(A) – повертає визначник матриці.

A' – транспонування матриці.

trace(A) – слід матриці (сума діагоналі).

inv(A) – $A^{-1} = E/M$ – отримання зворотної матриці

diag(n:m) – створення квадратної матриці з діагоналлю від n до m через 1

Візуалізація обчислень

plot(x, y, s) — побудова графіку функції $y = f(x)$, вектор констант s не обов'язковий, і використовується для задавання параметрів стилю графіку. Може містити три символи: «Тип точки», «Колір лінії», «Тип лінії» згідно табл. А.2 та мати наприклад таку форму запису: $'+r'$

plot(x1, y1, s1, x2, y2, s2, ... xn, yn, sn) — побудова великої кількості функцій на одному графіку.

stairs(x, y, s) — виводить графік елементів одновимірного масиву у вигляді ступінчатої функції.

stem (*x,y,s*) — виводить графік елементів одновимірного масиву у вигляді вертикальних ліній, які закінчуються в точках графіку та помічаються кружечками.

hist (*y,m*) — побудова гістограми, де *y* – вектор чисел, *m* – кількість інтервалів.

ezplot(*f, x_n, x_k*) — дозволяє побудувати графік функції, якщо вона задана в символьному вигляді (*f*='*x*²+1'), де *x_n* – початкове значення *x*, *x_k* – кінцеве значення *x*.

title('Назва графіку'), *xlabel*('назва осі *x*'), *ylabel*('назва осі *y*') — написи на графіку. Наносяться після побудови самої функції.

grit on/off — ввімкнення/вимкнення сітки

hold on/off — ввімкнення/вимкнення функції виведення графіку в одне і те ж вікно.

loglogx(...), *semilogx* (...), *semilogy* (...) — побудова графіків в логарифмічному масштабі. *loglogx*(...) – всі осі в логарифмічному масштабі, *semilogx* (...) – тільки вісь *x*, *semilogy* (...) – тільки вісь *y*. Синтаксис аналогічний функції *plot* (*x,y,s*)

Таблиця А.2 Стили графіків

Тип точки		Колір лінії		Тип лінії	
.	Крапка	Y	Жовтий	-	Суцільна
o	Коло	M	Фіолетовий	:	Подвійний пунктир
x	Хрест	C	Голубий	-.	Штрихпунктир
+	Плюс	R	Червоний	--	Штрих
*	Зірочка	G	Зелений		
s	Квадрат	B	Синій		
D	Ромб	W	Білий		
v, ^, <, >	Трикутник	K	Чорний		
P	П'ятикутник				
H	Шестикутник				

polar (*Θ, r, s*) — побудова графіків в полярній системі координат.

axis([*XMIN XMAX YMIN YMAX*]) — функція, яка задає діапазон по осям, в яких потрібно відобразити графік. Виконується після команди побудови графіків

subplot(*m, n, p*) — розбиває область виведення графіка на *m* х *n* частин, а *p* вказує в яку саме частину виводити графік.

ДОДАТОК Б. ПРИКЛАД ОФОРМЛЕННЯ ЛАБОРАТОРНОЇ РОБОТИ

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»
Радіотехнічний факультет
Кафедра радіоконструювання та виробництва радіоапаратури

Лабораторна робота №1
з курсу «Автоматизація обробки технічної інформації»
«Візуалізація результатів розрахунків»

Перевірив
асистент каф. КіВРА
Адаменко В.О.

« __ » _____ 2016р.

Виконали студенти

гр. РІ-__

Іванов І.В.

Гриць П.А.

« __ » _____ 2016р.

Київ – 2016р.

Мета роботи: Ознайомитися з середовищем моделювання MATLAB, особливостями створення та відображення сигналів.

Лістинг програми:

```
% Очищення командного вікна; закриття графічних вікон;  
видалення змінних  
clc; close all; clear;  
  
% Кількість точок розрахунку  
N = 1000;  
% Частота сигналу  
f0 = 1e6;  
% Амплітуда сигналу  
Um = 5.1;  
% Фаза сигналу  
phi = 0;  
  
% Генерування сигналу  
t = (0:pi/N:pi)*1e-6; % Вектор часу в мікросекундах;  
y = Um*sin(2*pi*f0*t+phi); % Досліджуваний сигнал;  
  
% Дискретизація сигналу  
% Для проведення дискретизації сигналу візьмемо кожен 20 точку  
масиву y  
k_d = 20; % коефіцієнт часового прорідження  
yd = y(1:k_d:end);  
  
%Демонстрація квантування дискретного сигналу з кроком  
квантування 1В  
  
yk=fix(yd);  
  
% Виведення графіків на екран  
figure (1) % створення першого вікна для виведення графіку  
subplot(2,1,1) % розбиття вікна на рядки та стовпчики 2x1, та  
виведення графіку в першу чарунку  
  
plot (t*1e6, y) %виведення графіку неперервного сигналу, вісь  
x в мікросекундах  
grid on % включення відображення сітки  
title ('Синусоїдальне коливання в часовій області') % назва  
графіку  
ylabel ('Um, В') % підпис осі y  
xlabel ('t, мкс') % підпис осі x  
axis ([0 pi 1.1*min(y) 1.1*max(y)]) % розмір області  
відображення [xmin xmax ymin ymax]  
  
subplot(2,1,2)  
plot (t*1e6, y, '--g') % виведення графіку початкового сигналу
```



```

hold on % ввімкнення накладення графіків
stem (t(1:k_d:end)*1e6, yd, '*r') %виведення графіку
дискретного сигналу, вісь x в мікросекундах
stairs (t(1:k_d:end)*1e6, yk) %виведення графіку квантованого
сигналу, вісь x в мікросекундах

legend ({'Початковий сигнал', 'Дискретний
сигнал', 'Квантований сигнал'}, 'location', 'SouthEast')%
виведення легенди

grid on
title ('Дискретизований сигнал та квантований сигнал')
ylabel ('Um, В')
xlabel ('t, мкс')
axis ([0 pi 1.1*min(y) 1.1*max(y)])

```

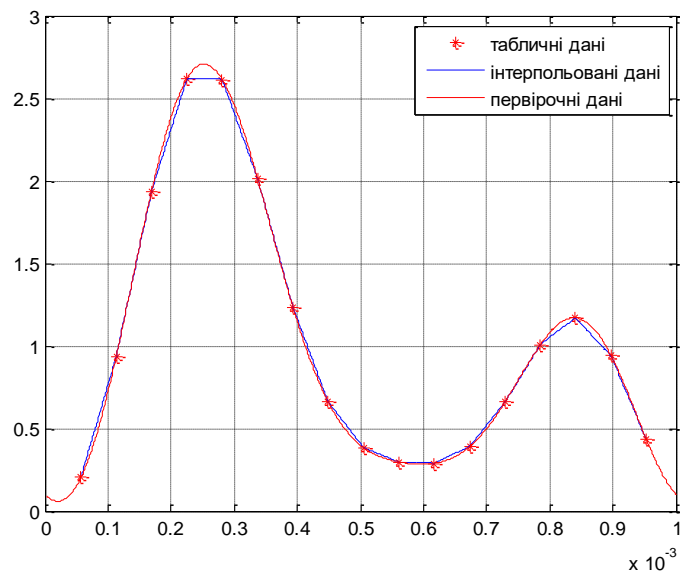


Рисунок 1 — Сплайнова інтерполяція

Висновки: Проведено моделювання неперервного, дискретного та квантованого сигналу в середовищі MatLAB. Досліджено особливості виведення різних типів сигналів у вигляді графіків.